Universidade Federal de Alagoas (UFAL)
Campus Arapiraca

Programação Orientada a Objetos (POO)

08 - Estruturas de dados (Arrays e Coleções)

Alexandre de Andrade Barbosa

alexandre.barbosa@arapiraca.ufal.br

Objetivos

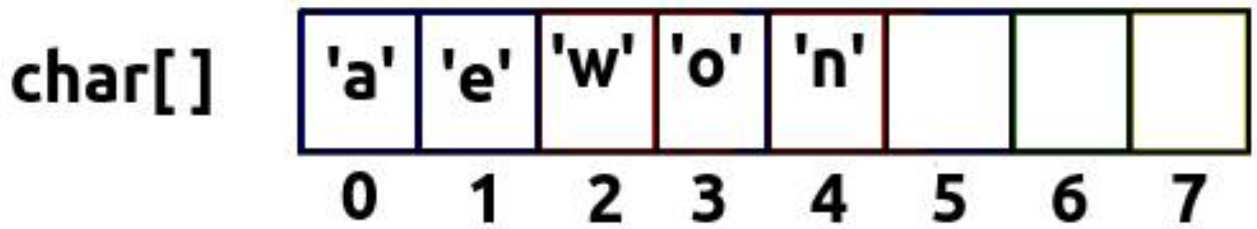
Objetivos

- Descrever o que são arrays e apresentar como utilizar esta estrutura
- Descrever a API de coleções em Java e exemplificar três tipos de coleções

Estruturas de dados (Arrays e Coleções)

Array

- Um array é uma estrutura de dados de **tamanho fixo** que pode armazenar elementos de um **mesmo tipo**
- Os elementos de um array podem ser acessados diretamente se sua posição for conhecida
- É possível inserir, manipular e excluir qualquer elemento utilizando o índice correspondente



Array de char de tamanho 8

Estruturas de dados (Arrays e Coleções)

Array

```java
public class ArrayExemplo01 {
    public static void main(String[] args) {
        // declaracao de um array de inteiros
        int[] inteiros;
        // inicialização do array, determinado seu tamanho
            (imutável)
        inteiros = new int[5];
        // inserção do valor 5 na posição 0 (primeira posição)
        inteiros[0] = 5;
        // inserção do valor 3 na posição 4 (última posição)
        inteiros[4] = 3;
        // impressão do valor armazenado na posição 0
        System.out.println(inteiros[0]);
        // nova inserção na posição 0
        inteiros[0] = 2;
        // impressão do valor armazenado na posição 0
        System.out.println(inteiros[0]);
    }
}
```

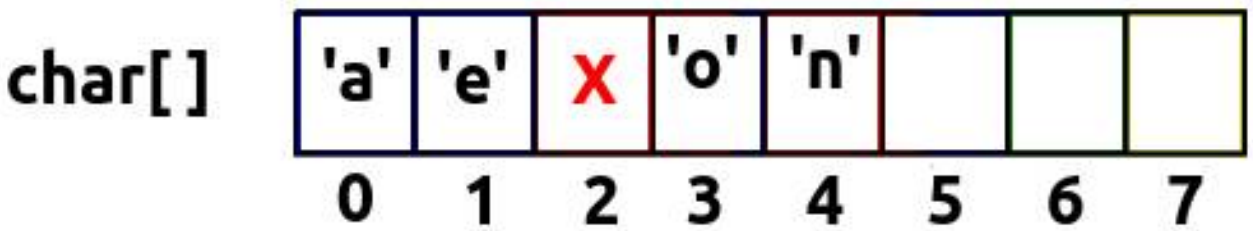
Estruturas de dados (Arrays e Coleções)

Array

```java
public class ArrayExemplo02 {
    public static void main(String[] args) {
        // declaracao de um array de inteiros
        int[] inteiros;
        // inicialização do array, determinado seu tamanho
            (imutável)
        inteiros = new int[5];
        // iterando sobre cada posição do array
        for (int i = 0; i < inteiros.length; i++) {
            // inserção do valor i, na posição i
            inteiros[i] = i;
        }
        // iterando sobre cada posição do array
        for (int i = 0; i < inteiros.length; i++) {
            // impressão do valor armazenado na posição i
            System.out.println(inteiros[i]);
        }
    }
}
```

Estruturas de dados (Arrays e Coleções)

Array



Array de char onde o elemento da posição 2 foi removido

Estruturas de dados (Arrays e Coleções)

Exercícios resolvidos

Exercício

1. Crie uma programa em Java que permita a inserção de números em um Array de tamanho 5 e determine qual foi:
   - (a) o maior valor
   - (b) o menor valor
   - (c) a média dos valores

Estruturas de dados (Arrays e Coleções)

Coleções

- As coleções em Java são classes que implementam uma vasta gama de estruturas de dados
- A API de coleções (ou *collections framework*) divide as estruturas em:
  - Listas (List) - coleção de tamanho variável de elementos de um mesmo tipo, permite inserção de elementos duplicados e possui ordem específica entre os elementos;
  - Conjuntos (Set) - coleção de tamanho variável de elementos de um mesmo tipo, não permite inserção de elementos duplicados (tal como em um conjunto matemático) e não possui ordem entre os elementos;
  - Mapas (Map) - coleção de elementos que possuem um objeto chave e um objeto valor associados (tal como dicionários em Python)
- Toda coleção Java possui métodos para inserir e remover um elemento

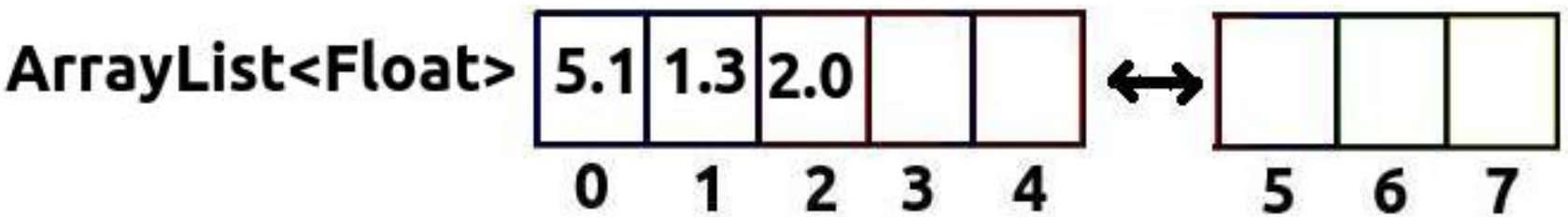
Estruturas de dados (Arrays e Coleções)

Coleções: List: ArrayList

- Um tipo de lista existente na API de coleções Java é o ArrayList
- Um ArrayList utiliza um array internamente para armazenar os elementos
- O tamanho do ArrayList pode ser alterado após sua criação
- Diversos métodos são fornecidos na interface de ArrayList:
  - *add(?)* permite a inserção de um elemento
  - *get(int)* permite a recuperação de um elemento
  - *set(int, int)* permite a substituição de um elemento
  - *remove(int)* permite a remoção de um elemento

Estruturas de dados (Arrays e Coleções)

Coleções: List: ArrayList



ArrayList de Float com três elementso inseridos

Estruturas de dados (Arrays e Coleções)

Coleções: List: ArrayList

```
1  import java.util.ArrayList;
2
3  public class ArrayListExemplo01 {
4
5      public static void main(String[] args) {
6          // declaracao de um ArrayList de inteiros
7          ArrayList<Integer> inteiros;
8          // inicialização do ArrayList
9          inteiros = new ArrayList<Integer>();
10         // inserção do valor 5
11         inteiros.add(5);
12         // inserção do valor 3
13         inteiros.add(3);
14         // impressão do valor armazenado na posição 0
15         System.out.println(inteiros.get(0));
16         // nova inserção na posição 0
17         inteiros.set(0, 2);
18         // impressão do valor armazenado na posição 0
19         System.out.println(inteiros.get(0));
20     }
21
```

Estruturas de dados (Arrays e Coleções)

Coleções: List: ArrayList

```
1  import java.util.ArrayList;
2
3  public class ArrayListExemplo02 {
4
5      public static void main(String[] args) {
6          ArrayList<Integer> inteiros = new ArrayList<Integer>();
7          inteiros.add(5);
8          inteiros.add(3);
9          inteiros.add(9);
10         for (int i = 0; i < inteiros.size(); i++) {
11             System.out.println(inteiros.get(i));
12         }
13         for (Integer integer : inteiros) {
14             System.out.println(integer);
15         }
16     }
17
18 }
```

Estruturas de dados (Arrays e Coleções)

Exercícios resolvidos

Exercício

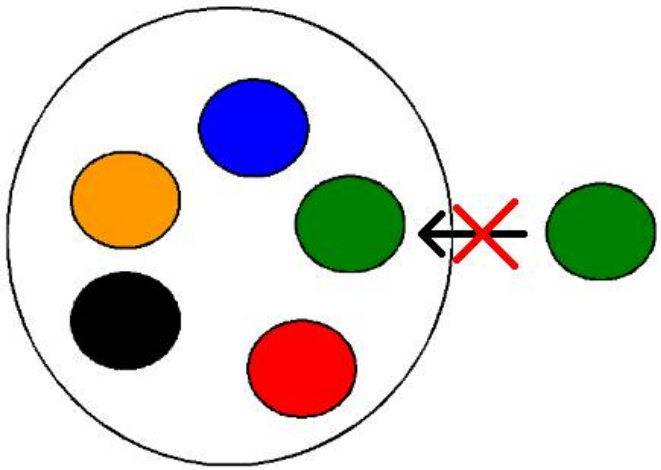
1. Crie uma programa em Java que permita a inserção de números em um ArrayList e determine qual foi:

   Obs.: A inserção só deve parar quando o usuário solicitar

   - (a) o maior valor
   - (b) o menor valor
   - (c) a média dos valores

Estruturas de dados (Arrays e Coleções)

Coleções: Set



Conjunto com cinco elementos.

O elemento externo (verde) não pode ser inserido pois já existe no conjunto

Estruturas de dados (Arrays e Coleções)

Coleções: Set: HashSet

```
1  import java.util.HashSet;
2
3  public class SetExemplo01 {
4
5      public static void main(String[] args) {
6          // declaracao de um HashSet de inteiros
7          HashSet<Integer> inteiros;
8          // inicialização do HashSet
9          inteiros = new HashSet<Integer>();
10         // inserção do valor 5
11         inteiros.add(5);
12         // inserção do valor 3
13         inteiros.add(3);
14         // inserção do valor 5 (repetido??)
15         inteiros.add(5);
16         // impressão de todos os valores armazenados
17         System.out.println(inteiros);
18     }
19
20 }
```

Estruturas de dados (Arrays e Coleções)

Coleções: Set: HashSet

```
1  import java.util.HashSet;
2
3  public class SetExemplo02 {
4
5      public static void main(String[] args) {
6          HashSet<Integer> inteiros = new HashSet<Integer>();
7          inteiros.add(5);
8          inteiros.add(3);
9          inteiros.add(9);
10         for (Integer integer : inteiros) {
11             System.out.println(integer);
12         }
13     }
14
15 }
```

Estruturas de dados (Arrays e Coleções)

Exercícios resolvidos

Exercício

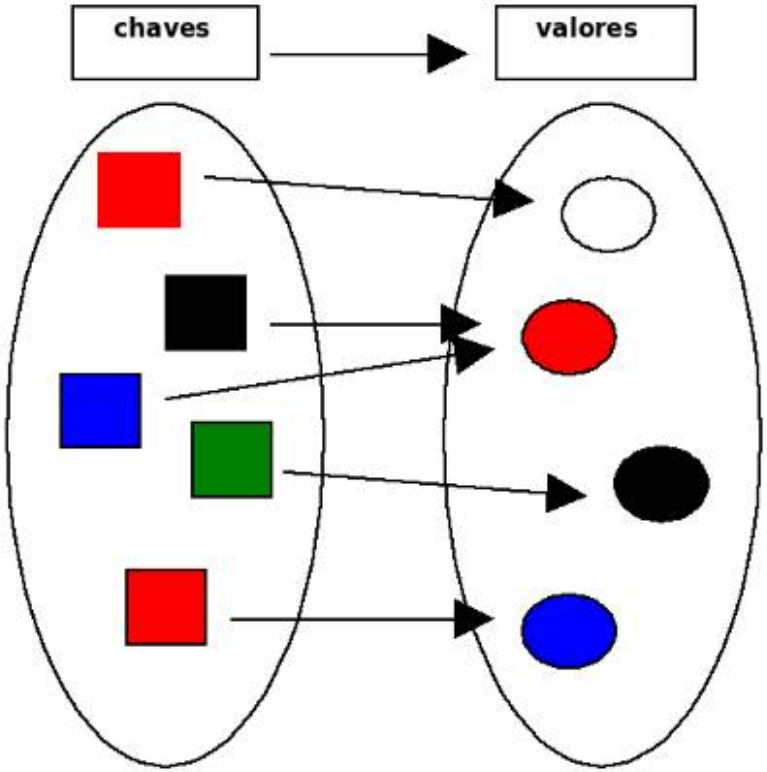
1. Crie uma programa em Java que permita a inserção de números em um HashSet e determine qual foi:

   Obs.: A inserção só deve parar quando o usuário solicitar

   - (a) o maior valor
   - (b) o menor valor
   - (c) a média dos valores

Estruturas de dados (Arrays e Coleções)

Coleções: Map



Mapa com cinco elementos chave e quatro elemntos valor.

Estruturas de dados (Arrays e Coleções)

Coleções: Map

```
1  public class HashMapExemplo01 {
2
3      public static void main(String[] args) {
4          // declaracao de um HashMap de chaves inteiras e
               valores
5          HashMap<Integer, Integer> inteiros;
6          // inicialização do HashMap
7          inteiros = new HashMap<Integer, Integer>();
8          // inserção da chave 3 com valor 5
9          inteiros.put(3, 5);
10         // inserção da chave 2 com valor 4
11         inteiros.put(2, 4);
12         // impressão do HashMap
13         System.out.println(inteiros);
14         // substituição do valor associado a chave 2
15         inteiros.put(2, 6);
16         // impressão do HashMap
17         System.out.println(inteiros);
18     }
19
20 }
```

Estruturas de dados (Arrays e Coleções)

Coleções: Map

```
1  import java.util.HashMap;
2
3  public class HashMapExemplo02 {
4
5      public static void main(String[] args) {
6          HashMap<Integer, Integer> inteiros = new
               HashMap<Integer, Integer>();
7          inteiros.put(3, 5);
8          inteiros.put(2, 9);
9          inteiros.put(1, 3);
10         for (Integer key : inteiros.keySet()) {
11             System.out.println(key);
12         }
13         for (Integer value : inteiros.values()) {
14             System.out.println(value);
15         }
16         for (Integer key : inteiros.keySet()) {
17             System.out.println(key + ": " + inteiros.get(key));
18         }
19     }
20
```

Estruturas de dados (Arrays e Coleções)

Exercícios resolvidos

Exercício

1. Crie uma programa em Java que permita a inserção de pares "Nome:idade" em um HashMap, e ao final da inserção determine: **Obs.:** A inserção só deve parar quando o usuário solicitar
   - (a) quem é a pessoa mais velha
   - (b) quem é a pessoa mais jovem
   - (c) a media de idade

Estruturas de dados (Arrays e Coleções)

Exercícios

Exercício

1. Crie uma aplicação Java para armazenar nomes e notas de alunos. Após fornecer um nome e uma nota para cada um, deve-se fornecer: (a) o nome do aluno com a maior nota (b) o nome de todos os alunos com nota menor que 5,0. Qual a melhor estrutura para utilizar na implementação deste programa? Por que?
2. Crie uma aplicação Java para armazenar telefones (apenas os números de telefone importam). Não deve ser possível inserir um número de forma duplicada. Qual a melhor estrutura para utilizar na implementação deste programa? Por que?
3. Crie uma aplicação Java que recebe cinco valores inteiros, ordena os valores e permite realizar uma busca. Qual a melhor estrutura para utilizar na implementação deste programa? Por que?

Estruturas de dados (Arrays e Coleções)

Exercícios

Exercício

1. Utilizando as classes conta-conta corrente e conta poupança, implemente a classe Cliente de modo que este possua diversas contas e uma operação para listar suas contas, a qual em sua execução delega a exibição dos dados (número e saldo) para a conta correspondente
2. Utilizando as classes conta-conta corrente e conta poupança, implemente uma classe 'Histórico' que deve ser associada a uma conta. O 'Histórico' de uma conta armazena todas as operações (depósito e saque) com seus respectivos valores e datas.
3. Utilizando as classes Conta e Cliente implementadas, crie uma classe 'Agência' a qual deve possuir Clientes e Contas

Estruturas de dados (Arrays e Coleções)

Resumo

Resumo

- Um array é uma estrutura de dados de **tamanho fixo** que pode armazenar elementos de um **mesmo tipo**
- As coleções em Java são classes que implementam uma vasta gama de estruturas de dados
- A API de coleções Java é dividida em Listas, Mapas e Conjuntos

Leituras recomendadas

FURGERI, S.
Java 6 - Ensino Didático
Capítulo 8: Utilização de Arrays

Caelum
Java e Orientação a Objetos, 2011.
Capítulo 05: Um pouco de Arrays
`http://www.caelum.com.br/apostila-java-orientacao-objetos/collections-framework/`

Estruturas de dados (Arrays e Coleções)

Perguntas?

Alexandre de Andrade Barbosa
alexandre.barbosa@arapiraca.ufal.br